Chapter 20

By:

ETISHA JAIN

VISHU GOYAL

Linux 20.1 Understanding SSH Key-based Login

. Passwords have one disadvantage they can be guessed so for that reason SSH key-based login it's much more secure.
. we use ssh-keygen
. It is going to generate two keys
. There is a private key and there is a public key.
. The private key is stored in a secure location .
. The public key must be copied over to your home directory on server and we will use ssh-copy-id command and once it is copied then if we are using ssh command to connect to server then our server will look up your public key then our server will look up your public key and based on it will send an answer and this answer is packet that is encrypted with your public key and we are going to launch a private key against that packet and if private key matches the public key then we can send back answer (an encrypted answer) . Mathematically public and private are related.

Linux 20.2 Setting up SSH key-based login

. ssh-keygen creates a public/private key pair for current user
    . Setting a passphrase for the private key makes it more secure, but less convenient.
. ssh-copy-id copies the public key over to the target server
. ssh-agent /bin/bash allocates space in the bash shell to cache the private key passphrase
. ssh-add adds the current passphrase to the cache

Practical
- ssh-keygen
By default private key is written to the directory .ssh in the current user home directory and there will be file id_rsa .
If press enter it will use empty passphrase
- ssh-copy-id 192.168.4.237(IP address or name of target server)
- ssh 192.168.4.237
In text control -- ctrl+alt+f3
- root
- ssh-agent /bin/bash
- ssh-add
- ssh 192.168.4.235
- exit

Linux 20.3 Changing Common SSH Server options

. Server options are set in /etc/ssh/sshd_config
. Client options can be set in /etc/ssh/ssh_config
    . Port 22(need to configure SE Linux to allow changed port as well)
    . PermitRootLogin (switch root login off)
    . PubkeyAuthentication (by default is on and should be on)
    . PasswordAuthentication (allow users to authenticate using a password)
    . X11Forwarding (allows users to forward a graphical screen to another server)

Practical
- vim /etc/ssh/sshd_config
ListenAddress listens to ip address 0.0.0.0 by default
ListenAddress 192.168.4.237
PermitRootLogin no
AllowUsers student
PasswordAuthentication (if we disable this then only users that have a valid public private key pair are allowed to log in .
- systemctl restart sshd
- ssh root@localhost

Linux 20.4 Securely Copying Files

. scp can be used to securely copy files over the network , using the sshd process.
. scp file1 file2 student@remoteserver:/home/student
. scp -r root@remoteserver:/tmp/files . (-r for recursive it will reach out to remote server and will copy contents of tmp files to current directory)
. sftp offers an FTP client interface to securely transfer files using SSH
. Use put /my/file to upload a file to remote server(/my/file is in local system)
. Use get /your/file to download a file to current directory
. Use exit to close an sftp session.

Practical
- scp /etc/hosts linda@192.168.4.210:/tmp/
- mkdir temp
- scp -r root@192.168.4.210:/etc temp/
- cd temp/
- ls
- cd ..
- sftp 192.168.4.210
>help
>lpwd
>pwd
>cd /
>pwd
>ls
>cd etc
>lpwd
> lcd /tmp
> lpwd
> get /etc/passwd
>pwd
>put /etc/passwd
>exit
vsftpd -very secure ftp daemon sftp is much more secure becoz it's using SSH so preferred.

Linux 20.5 Securely Synchronizing Files

. rsync is using ssh to synchronize files.
. If source and target file already exists , rsync will only synchronize their differences
. The rsync command can be used with many options of which the following are most common
    . -r will recursively synchronize the entire directory tree
    . -l will synchronize symbolic links
    . -p preserves symbolic links
    . -n will do a dry run before actually synchronizing
    . -a uses archive mode, which js equivalent to -rlptgoD (almost preserve everything)
    . -A uses archive mode and also synchronizes ACLs
    . -X will synchronize SELinux context as well

Practical
- ssh 192.168.4.210
- touch /etc/mynewfile{1..10}
- ls -l /etc/my*
- exit
- rsync -ar root@192.168.4.210:/etc/ temp/
- a for archieve
- r for recursive
- ls temp/mynewfile*

THANK YOU